An XML Object Model for Rich Vehicle Routing Problems

Kevin Sim*, Emma Hart, Neil Urquhart
Centre for Algorithms, Visualisation and Evolving Systems
Edinburgh Napier University, Merchiston Campus, Edinburgh, EH10 5DT
Email: {k.sim, e.hart. n.urquhart}@napier.ac.uk*

Tim Pigden
Optrak Distribution Software Ltd
Suite 6, The Maltings, Hoe Lane, Ware, SG12 9LR
Email: tim.pigden@optrak.com

May 2015

Summary
We describe a new rich VRP model that captures many real-world constraints, following a recently proposed taxomomy that addresses both scenario and problem physical characteristics. The model is used to generate 4800 new instances of rich VRPs which is made freely available. To the best of our knowledge this represents the most comprehensive resource of rich VRP problems available, and provides a platform for researchers to conduct rigorous comparisons of new methods and solvers, moving academic research much closer to real practice in the future.

Keywords: *Rich Vehicle Routing problem, model, benchmark datasets*

1 Introduction

More than half of century since Dantzig and Ramser [1] first introduced the Vehicle Routing Problem (VRP) in 1959, academics and practitioners continue to actively explore variants of this problem and introduce new methods to provide solutions. A survey in 2009 by Laporte [2] summarised the state-of-the-art in research into the classical VRP charting the developments in exact methods and heuristics, noting that heuristics now enable realistically sized instances of VRP to be solved adequately. However, the author points out the gap between classical VRP models and the stochastic and dynamic features of problems that are apparent in many real world problems. Caceres-Cruz *et al* [3] go further in recognising that in addition to stochastic and dynamic features, real VRP problems deal with multi-objective optimisation functions and a wide variety of constraints that cover factors such as heterogeneous fleet, time factors relating to congestion, the need to combine routing with driver scheduling and the increasing need to provide compliance with environmental regulations. In response to this, the term *rich VRP* has been coined to describe VRP instances that account for some or all of these factors.

Much research has been published under this heading: two recent surveys [3, 4] provide a detailed summary of the state-of-the art including problem combinations, constraints defined, and approaches found, demonstrating the rich variety of problems now being tackled. However, we note that unlike in the classical VRP literature, it is difficult to find *datasets* containing sufficient benchmark instances to fairly evaluate new approaches; many of the published works refer to real-world problems where only a few instances are described and often datasets are not published. To address this issue, we describe a new model of a rich VRP that encapsulates many real constraints, informed by input from practitioners[1]. The model is instantiated in an extensible software-framework that is straightforward to extend with additional constraints. Using the framework, we generate a significant resource of 4800 problems that are made freely available. Instances cover a wide range of sizes and are generated on a map of the UK; driving distances and times are generated from mapping software utilising actual road-networks and thus potentially includes asymmetric distances

[1]Optrak Vehicle Routing Software www. optrak.com

between points. The instances provide a foundation for rigorous comparisons of algorithms and methods, that it is hoped will drive forward academic research within this area in the future.

In the next section we provide some background to the RVRP, grounding our model in a recently proposed taxonomy for RVRP. The software framework and problem instances are then described. All instances are downloadable from [5].

2 Background

Motivated by the need to address the gap that is clearly apparent between the complex characteristics of real-life VRPs and much academic research, increasingly more complex variants of the classical VRP problem are tackled, often referred to as *Rich VRP* problems. However, Lahyani *et al* recently observed that across the wide-range of research published under this heading, there was no precise definiton of what criteria led to a problem being classified as *rich*, and proposed a generic taxonomy for RVRP. The taxonomy differentiates between Scenario Characteristics and Problem Characteristics; within each class, subclasses describe characteristics at the *strategic*, *tactical* and *operational* levels within a hierarchical structure. The taxonomy is shown in table 1 (note that no characteristics at the operational level are described for Scenarios).

Based on this taxonomy and an extensive analysis of literature, Lahaynai *et al* propose a precise definition of a VRP, given below:

Definition 1 *RVRP extends the academic variants of the VRP in the different decision levels by considering at least four strategic and tactical aspects in the distribution system and including at least six different daily restrictions related to the physical characteristics. When a VRP is mainly defined through strategic and tactical aspects, at least five of them are present in a RVRP. When a VRP is mainly defined through physical characteristics, at least nine of them are present in a RVRP.*

The authors note that under this definition, many papers previously considered as 'rich' may not comply. Caceres-Crux *et al* note that in fact, the RVRP definition evolves continuously and conclude that "*the RVRP reflects, as a model, most of the relevant attributes of a real-life vehicle-routing distribution system*", and therefore should be seem as an accurate representation of a real-life distribution system, and hence methods for solving instances characterised as RVRP should be directly applicable to a real-life scenario.

In order for research in developing new methods that deal with the complex set of constraints associated with RVRP to progress, it is beneficial for the community to have access to a resource of relevant benchmark problems on which approaches can be rigorously compared. However, unlike in classical VRP in which many datasets are well known [6], an equivalent resource of RVRP problems is not readily available. For example, Lahyani *et al* [4] survey 16 recent papers relating to pure vehicle routing that describe rich problems. Of these 15 of them focus on real world problems, each with very specific constraints, and hence do not provide a wide ranging suite of benchmarks that enable algorithms to be easily and fairly compared. On the other hand, the non-rich VRP literature abounds with benchmarks datasets. We briefly examine these before proposing a new rich VRP model that is easily extensible, and a freely available problem generator that can be used to generate multiple instances

2.1 Benchmark Data for VRP

A number of web-based resources provided aggregated collections of VRP datasets. The VRP-REP project[6] aims to provide the VRP community with a collaborative open data platform, enabling users to share instance files, check solutions and track solutions. It currently provides 34 datasets, the largest of which contains 180 instances. The majority of these datasets do not meet the definition of RVRP, although the newer additions move towards this (for example Mendoza *et al* [7] provide a dataset that models stochastic demands and duration constraints). The Networking and Emerging Optimization group [8] provide datasets for capacitated VRP, including those with time-windows and pickups and deliveries, as well as multi-depot and periodic problems. More recently, Goeke and Schneider [9] provide instances for a variant of VRP that includes electric vehicles, time-windows and recharging constraints.

Although many distinct data-sets are available, the relatively small size of each set limits the extent to which rigorous algorithms comparisons can be made, particularly in research using approximation methods that can deal with large instances and find solutions in relatively short amounts of time. A specific case of this is recent research in *hyper-heuristics* which deals with methods that find cheap but acceptable solutions

Table 1: Rich VRP Taxonomy from [4]: characteristics reflected in the proposed model are shown in *italics*

(a) Scenario Characteristics

Strategic	**Tactical**
Input Data	*Static*
	Dynamic
	Deterministic
	Stochastic
Decision Management Options	Routing
	Inventory and Routing
	Location and Routing
	Routing and Driver Scheduling
	Production/Distribution Planning
Number of Depots	*Single*
	Multiple
Operation Type	Pickup or *Delivery*
	Pickup and Delivery
	Backhauls
	Dial-a-ride
Load Splitting Constraints	*Splitting Allowed*
	Splitting Not Allowed
Planning Period	*Single Period*
	Multi-period
Multiple use of vehicles	Single Trip
	Multi-Trip

(b) Physical Characteristics

Strategic	**Tactical**	**Operational**
Vehicles	Type	Homogeneous
		Heterogeneous
	Number	*Fixed*
		Unlimited
	Structure	*Compartmentalised*
		Not compartmentalised
	Loading Policy	chronological
		no order
	Capacity Constraints	
Time Constraints	*Restriction on customer*	
	Restriction on road access	
	Restriction on depot	
	Service time	
	Waiting time	
	Driver regulations	
Time Window Structure	*Single time window*	
	Multiple time windows	
Incompatibility Constraints	*Equipment*	
	Compartment restrictions	
Specific Constraints		
Objective Function	Single Objective	
	Multi-objective	

that generalise over very large problem sets and are often tested with problem databases containing 1000s of instances.

Recognising the dual need for new benchmarks that capture rich vehicle routing problems that reflect real-world constraints and diverse collections of benchmarks to encourage comparative research, we propose a new RVRP model. The model is translated to an object-oriented software framework that is easily extensible, therefore new scenario and problem physical constraints can be added in future. We provide 4800 problem instances of varying size and complexity for future research. The model is described in detail in the next section.

3 Model

The proposed model aligns with the taxonomy proposed by Lahyani[4] and shown in table 1. In this table, features of the new model are shown in italics within the taxonomy. Note that the model contains one new characteristic at the *strategic* level, in the use of real road-network data to calculate distances and times.

3.1 Scenario Characteristics

We describe a *static* problem p given by the tuple $p = \{J, V, d, M, s, E\}$ where J is a set of jobs to be delivered, V is a set of vehicles, d defines a *single* depot location and M is a set of 2 matrices giving the asymmetrical distances and times between all locations. Only *Routing* is considered, i.e. it is assumed that a driver will be available for a particular vehicle at all times.

Planning is over a single continuous time period given by s. *Multi-trips* by the same vehicle are allowed in a day.

The remaining term E defines the time constraints imposed on the driver and are described in Section 3.2.2.

We consider *delivery* problems only. Each job $j_i \in J$ is described by $j_i = \{l_i, U_i, C_i\}$. l_i gives the location of the customer the job is to be delivered to, U_i is the set of pallets of a product to be delivered. $C_i = \{w_i, t_i, q_i, \alpha_i, \beta_i, \gamma_i, \delta_i\}$ describe the set of constraints associated with the physical characteristics of a job and are described in sections 3.2.2 and 3.2.3. *Splitting* is allowed — the pallets associated with a job j_i can be delivered by multiple vehicles.

3.2 Scenario Characteristics

3.2.1 Vehicles

A *fixed* and *heterogeneous* vehicle fleet V is defined for each problem.

Each *compartmentalised* vehicle $v_j \in V$ is described by $v_j = \{T_j, e_j, q_j\}$ where T_j describes the set of temperature controlled compartments on the vehicle, e_j defines the vehicle's engine type and q_j indicates the availability of any additional equipment (tailgate) for the vehicle. Each compartment $t_k \in T_j$ is described by $t_k = \{r_k, n_k\}$ where r_k defines the compartment type and n_k indicates the capacity of the compartment. The values of r_k, n_k, e_j, q_j considered are listed below:

- $r_k \in \{FROZEN, REFRIGERATED, AMBIENT\}$
- $n_k \in \{6, 12, 24\}$
- $e_j \in \{DIESEL, HYBRID, GAS\}$
- $q_j \in \{TAILGATE, NONE\}$

Vehicles do not have a loading policy, that is, compartments can loaded and unloaded in any order.

3.2.2 Time constraints and time-windows

Each problem has employee constraints $E = \{x, b\}$ which define the maximum driving time allowed x before a compulsory break of length b is imposed. The set of physical constraints associated with a job j_i, given by $C_i = \{w_i, t_i, q_i, \alpha_i, \beta_i, \gamma_i, \delta_i\}$, are described in this and the following section. A single time-window for delivery w_i is assumed for each job (*customer restrictions*) during which the job should *arrive* at the customer location. In addition, each job has 4 terms defining *service times* $(\alpha_i, \beta_i, \gamma_i, \delta_i)$. α_i and β_i give the fixed loading time per job and loading time per pallet respectively at the depot . These terms are not applied for the first route assigned to a vehicle as this is generally completed prior to the service period. Similarly γ_i and δ_i define the fixed unloading time per job and unloading time per pallet at the customer's location. All durations are defined in minutes.

It is assumed that all vehicles are available for the duration of the planning period during which all routes must start and finish at the depot. When a vehicle (and its driver) return the depot, it immediately becomes available for reuse (the driver's cumulative driving time is reset).

3.2.3 Incompatibility Constraints

All pallets from a job have the same type $t_i \in \{FROZEN, REFRIGERATED, AMBIENT\}$ and must be placed in a compartment of the corresponding type. Some customers require that the vehicle that a load is delivered on must have particular equipment given by $q_i \in \{TAILGATE, NONE\}$.

3.2.4 Driving Times/Distances

The manner in which driving times are computed is not included in the RVRP taxonomy. We consider routing performed on a real road-network. Distances and times between any two locations are obtained from the open-source mapping product GraphHopper [10]. This returns *asymmetric* distances (and times) between any two points (e.g. accounting for one-way streets). Travel times are stored in the matrix m_t and distances in m_d. In the model supplied, the time between any two points is fixed and therefore does not account for rush-hour driving. The model could be extended to account for this using stepwise linear functions that then have to be combined in order to calculate (and minimise) travel times across the day.

3.2.5 Customer Locations

The depot and the customers are located at geographical locations derived from actual postcodes within the UK. In the model supplied, customer locations are selected at random from all locations listed in the UK postcode database that can be serviced within a maximum time-scale from the depot. This process ensures a realistic distribution of customer locations based on the actual distribution of residential and retail dwellings in the geographic area surrounding the depot.

3.2.6 Cost model

Vehicles incur different costs depending on their size and engine type. Three costs are considered; fixed, running and environmental costs. The problem model does not contain actual costs but contains 3 multipliers for each vehicle $v \in V$ which show the relative costs compared to a *standard* vehicle which we define as a *large* vehicle fitted with a *diesel* engine having one large ambient container of capacity *24* product units. Tables 2 and 3 show the values used to calculate the cost multipliers based on the vehicle's engine type and total capacity respectively. The 3 cost multipliers supplied with each vehicle are calculated by multiplying the values obtained from each of the two tables corresponding to the vehicles size and engine type.

Table 2: Cost multipliers corresponding to engine type

	Cost Multiplier		
Engine Type	*Fixed*	*Running*	*C02*
Diesel	1.00	1.00	1.00
Hybrid	1.07	0.80	0.80
Gas	1.04	0.70	0.95

Table 3: Cost multipliers corresponding to vehicle size

Vehicle Size (Capacity)	**Cost Multiplier**
Small (6)	0.35
Medium (12)	0.60
Large (24)	1.00

The actual costs associated with any vehicle can be calculated by scaling the costs incurred by a standard vehicle. Suggested costs for the standard vehicle type, based on current market prices, are given in table 4. The actual costs associated with any vehicle are described and calculated as follows.

Table 4: Costs associated with the standard vehicle type

Cost	**Value**
Fixed Cost	Âč242/day[11]
Running Cost	Âč0.36/KM[11]
Environmental Cost	Âč0.015/KM[12]

- **Fixed Costs** fc_j is the fixed cost associated with vehicle j and is calculated by multiplying the fixed cost multiplier of that vehicle by the standard vehicle cost per day. The standard vehicle fixed cost per day typically includes driver costs / depreciation, servicing, taxes and other costs that are independent of the distance covered by the vehicle.

- **Running Costs** dc_r is the running cost associated with route r which varies depending on the distance covered and the vehicle used. This is calculated as the standard vehicle running cost per km * distance covered by route * vehicle running cost multiplier. The standard vehicle cost per KM typically includes fuel, tyre wear and other distance dependent costs.
- **Environmental Costs** ec_r is the environmental cost associated with route r which varies depending on the distance covered and the vehicle used. This is calculated as the standard vehicle environmental cost per km * distance covered by route * vehicle environmental cost multiplier.

3.2.7 Objectives

Any single objective function can be defined as a sum over the costs described; alternatively, the problem can be treated as a multi-objective problem. An example single objective function that takes into consideration all the costs described as well as penalties incurred for breaking soft constraints is given by:

$$f(x) = \sum_{j=1}^{n} fc_j + \sum_{r\in R} dc_r + \sum_{r\in R} ec_r + Z_l + Z_o$$

The fixed cost fc_j is applied once for each vehicle used (any unused vehicle does not incur a cost). The total running cost $\sum_{r\in R} dc_r$ and environmental costs $\sum_{r\in R} ec_r$ are summed over all routes R conducted by all vehicles. $Z_l + Z_o$ are financial penalties imposed as described below.

- **Time Penalties** Z_l is the late/early penalty which is incurred for all pallets arriving at a customer's location outside of the specified time window. Each pallet arriving outside the time window incurs a fixed penalty of $(£1 \times time)^2$ where $time$ is the the number of minutes that the delivery is either early or late. Research on real world problems [] has shown that using a quadratic function to penalise broken time windows minimises the number of potential solutions that incur equal penalties.
- **Outsourcing Penalty** Z_o is the outsourced pallet penalty per pallet per KM. This is calculated using $arg_{max}\{£5/pallet/KM, £50/pallet\}$ where the distance is measured from the depot to the associated customers location. The second term defines a minimum cost for outsourcing a pallet.

Solutions can be evaluated using the suggested metric or may be evaluated using any user specified objective function. The model allows for relaxation of both hard and soft constraints as required. For example the end user could ignore suggested penalties for breaking soft constraints and disregard all costs focussing only on total distance, effectively turning the RVRP model into a conventional VRP. Hard constraints could also be ignored. For example jobs marked as having to be transported on an ambient container could be allowed to be delivered on any container type. This abstraction of the objective function allows the RVRP model implemented to be easily transformed using different scenarios.

4 Problem Generation

The section describes the process of generating the 4800 problem instances that are made available in XML format. For each problem $p = \{J, V, d, M, s, E\}$, the service period s during which all customers should be serviced is fixed to between 9am and 5pm on the 1st January 2015. The driver constraints $E = \{x, b\}$ representing the maximum continuous working time and break length are fixed for all problems at 270 minutes before which a 45 minute break is enforced such that the driver never works more than 270 minutes continuously. The break is applied either before (where unloading takes the cumulative working time over the threshold) or after a job is unloaded (where the next period of driving breaks the maximum working time). A driver returning to the depot has their cumulative working time reset. All instances generated have valid solutions that meet all hard and soft constraints without requiring any pallets to be outsourced. The fixed and variable loading and unloading times for jobs and pallets, given by the terms $\alpha_i, \beta_i, \gamma_i, \delta_i\}$, may vary and are defined in each problem instance. Note that loading times are not applied for jobs that are delivered on the first route by each vehicle.

Each problem instance is generated using a specific *vehicle fleet* and a specific *customer data set*. The procedure is described in the following sections.

4.1 Vehicle Fleet

Two parameters control the fleet of vehicles used — the size of the fleet vf_s and the vehicle class. The vehicle class describes a set of vehicles defined by the parameters $(containers, engine-type, equipment)$. Four classes of vehicle are considered:

- Vehicle class 1: Medium sized vehicle fitted with a diesel engine, no tailgate and a single ambient container. (1 possible vehicle type)
- Vehicle class 2. Medium or large vehicle with a single ambient container, no tailgate and any engine type. (6 possible combinations)
- Vehicle class 3: Medium or large vehicles of any engine type, with or without a tailgate and with either 1 ambient container or 1 refrigerated and 1 frozen container. (24 possible combinations)
- Vehicle class 4: All possible vehicle sizes, container combinations, engine types and equipment types (90 different combinations)

4.2 Customer Databases

Five sets of customers are used to generate problems. Each customer set has an associated distance and time matrix which are supplied separately to minimise file sizes. A single depot for all problem instances is used that is located in the city of Glasgow at a location roughly 7 KM from the city centre defined by the geographic coordinates (*latitude 55.853 longitude -4.309*).

A class of problems is generated by considering customers that can be serviced within a service-time rt_{max} minutes from the depot. We generate five classes of problems for $rt_{max} \in \{30, 60, 120, 240, 480\}$. Within each class, 1000 customers locations are randomly selected from the subset of postcodes that lie within rt_{max}; this ensures a realistic distribution of customer locations based on the actual distribution of residential and retail dwellings in the geographic area surrounding the depot.

Figure 1 shows the distribution of customers from 2 of the 5 customer data sets generated.

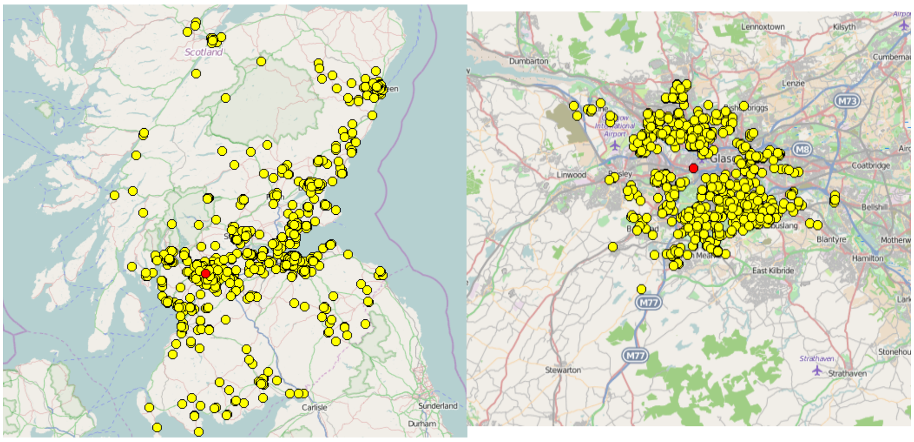

Figure 1: Geographical distribution of customers that can be serviced within 480 minutes and 30 minutes.

Distances (km) and travel times (rounded to the nearest minute) between all pairs of locations are then calculated and stored in matrices m_d, m_t respectively. These are derived using the Graphhopper library as described in section 3.2.4. The library makes use of *Dijkstra*'s algorithm to determine the shortest path between two locations, using a road network modelled as a directed graph. The time taken to travel between locations is correlated with the road type used with shorter travel times returned for higher capacity roads. A minimum driving time of 2 minutes is used where the software returns a shorter duration. The resulting data sets require *cleaning* due to inaccuracies present in the open source software which on occasion returns erroneous distance or time measurements. The resulting data sets are consequently reduced in size from 1000 to between 524 and 734 customers.

4.3 Generating Problem Instances

For each of the five customer sets, 10 problem instances are generated using all 96 combinations of the following parameters, resulting in 4800 problem instances in total:

- *Minimum round trip* : $rt_{min} \in \{0, rt_{max}/2\}$, where rt_{max} is the maximum round trip time used to generate the corresponding customer data set.
- *Vehicle Class* Four distinct classes (described in section 4.1).
- *Vehicle fleet size* $|V| \in \{4, 8, 16, 32\}$
- *Time Window* $\in 60, 120, 240$ minutes

For each instance, a fleet type is defined, and then v_{fs} vehicle types are randomly selected from the class. Having defined the customer set and a fleet, problem instances are generated by assigning route(s) to each vehicle using the process described by Algorithm 1.

Algorithm 1 Problem Generation

```
for all v ∈ V do
    Time Vehicle Utilised = 0
    repeat
        Add a Route for the current Vehicle
        repeat
            Select a customer at random that can be serviced within the remaining time (allowing time to return to the depot and allowing for any
            required driver breaks)
            Create a job for the customer with a random number of pallets between the limits (or the available capacity) of type (temperature) that
            corresponds to a randomly selected container with available capacity available to the vehicle
            Assign the time window that corresponds with the time that the vehicle will arrive at that location
        until Vehicle Capacity Reached OR Time Vehicle Utilised ≥ 450
    until Time Vehicle Utilised ≥ 450
end for
```

5 Benchmark Datasets

5.1 XML Description

The problems are released as XML files that reflect the model described in Section 3. The XML files comply to the Class Diagrams depicted in Figure 2 and Figure . In order to reduce file sizes, each problem instance is supplied without the required matrices that specify travel times and distances between all pairs of locations. The matrices associated with each company data set are supplied as *CompanyData* objects, also in XML format. Each of the 5 files is formatted as described by Figure 4 and contain the time and distance matrices that are required by a problem instance.

All problem instances have a unique integer ID ranging from 1000000 to 1004799. For each problem instance all jobs, vehicles, containers and pallets have a unique ID (ranging from 1...n). Customers can be identified by their unique ID or their location which remain the same where a customer appears in more than 1 of the data sets. A customer may require to have more than one job delivered for a single problem instance and therefore multiple jobs may exist where the associated customers have the same ID.

Problems are stored on the distribution site in a hierarchical directory structure that corresponds to the parameters that the problems were generated from. Each set of 10 problem instances that were generated using the same parameter combination are zipped in their own low level directory. The corresponding matrices can be identified using the *matrixID* given in each problem instance and can be located in the corresponding high level folder. The directory structure consists of 5 nested layers listed below from highest to lowest.

- **Company Data Set A :** $A \in \{1 \dots 5\}$ corresponding to $rt_{max} \in \{30, 60, 120, 240, 480\}$.
- **Minimum Trip Time B :** $B \in 0, 1$ corresponding to $rt_{min} \in \{0, rt_{max}/2\}$minimum round trip time
- **Vehicle Set Type C :** $C \in \{1 \dots 4\}$
- **Fleet Size D :** $D \in \{4, 8, 16, 32\}$

- **Time Window E :** $E \in \{60, 120, 240\}$

Within each of the lowest level folders is a single zip file containing the 10 problem instances generated using the same parameter settings. The associated distance and time matrices are accessible from the zipped XML document to be found in the corresponding high level folder.

5.2 Parsing problems using the supplied Java Parser

The distribution web site includes an XML parser, supplied as a jar file, that includes the problem model and a XMLParser Class containing a single method with the signature *getProblem(String instance, String companyDataSet)*. The method takes two strings as parameters representing the file names of a problem instance and the corresponding CompanyData object and returns a Problem object with the associated distance and time matrices added.

5.3 Other Formats or Programming Languages

For users wishing to use platforms other than Java, two XML schema definition files are supplied that allow the Problem and CompanyData structures to be recreated. The matrices required for a problem instance can be identified by the matrixId field of the Problem Class and by the ID field of the CompanyData Class.

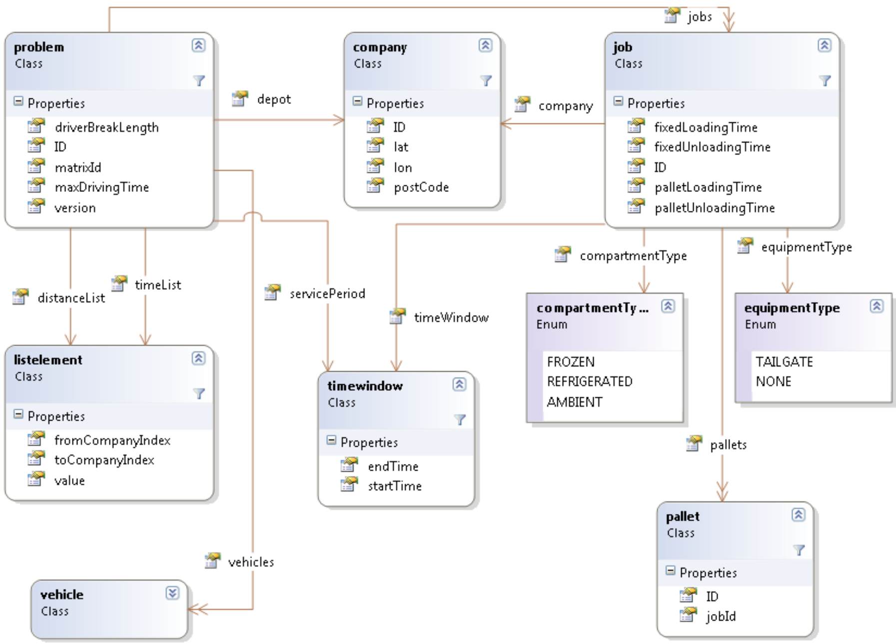

Figure 2: The Problem Model

6 Conclusion

It is clear that if research within vehicle routing is going to address the concerns of the industry, then large-scale instances and variants of the VRP that meet the definiton of *rich* problems are urgently required. We have proposed a new rich VRP model that incorporates load-splitting and multi-trip scenarios, combined with physical constraints that include heterogeneous fleets, multi-compartments, equipment and compartment constraints, time-window and multiple objectives. The model utilises time and distance data from a model of a real road-network, in which times are modified according to road capacity. The model is described using an XML format that can be read by any parser. 4800 instances are supplied, providing the biggest resource of rich VRP problems available to date. The framework can easily be extended to include further constraints to capture new characteristics of problems emerging in rapidly changing industry. From an academic perspective, the resource provides a means of rigorously evaluating new algorithms to assess performance across a diverse range of problems. This provides an essential mechanism for highlighting

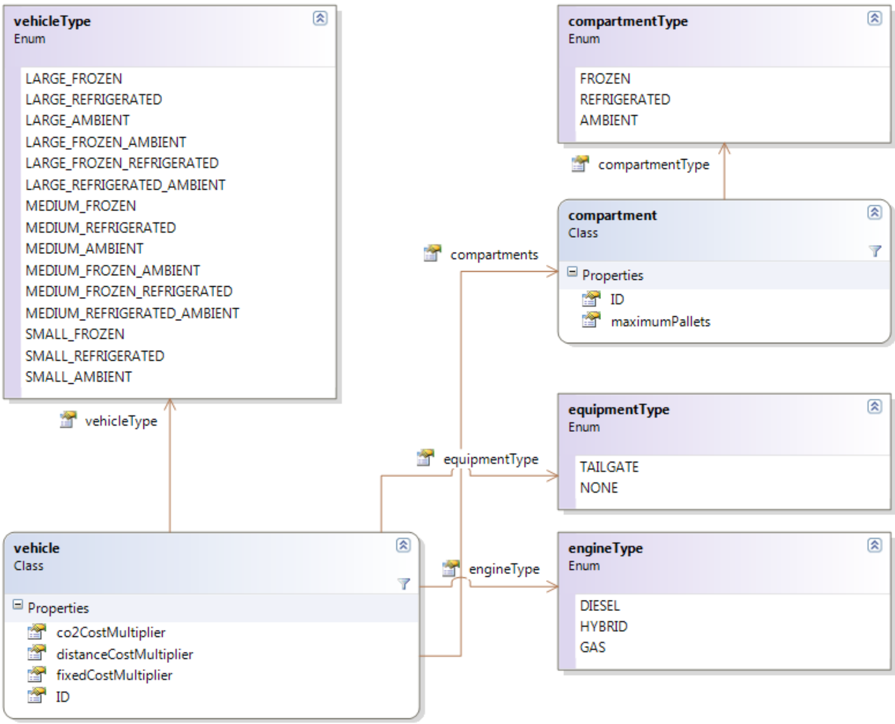

Figure 3: The Vehicle Model

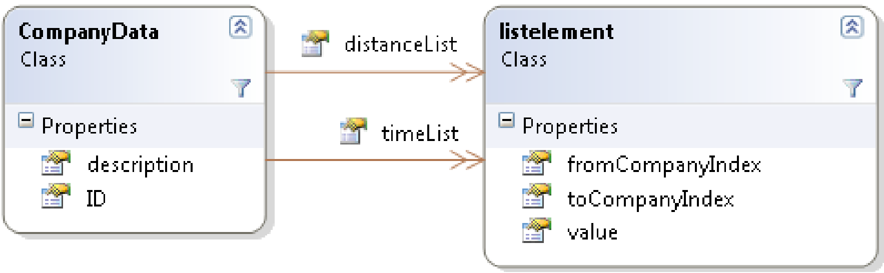

Figure 4: The Company Data Model

strengths and weaknesses of algorithms in relation to problem characteristics, rather than assuming a 'one-size-fits-all' approach, that will drive new innovation in the field.

References

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] G. Laporte, "Fifty years of vehicle routing," *Transportation Science*, vol. 4, pp. 408–416, 2009.

[3] J. Caceres-Cruz, P. Arias, D. Guimarans, D. Riera, and A. A. Juan, "Rich vehicle routing problem: Survey," *ACM Comput. Surv.*, vol. 47, pp. 32:1–32:28, Dec. 2014.

[4] R. Lahyani, M. Khemakhem, and F. Semet, "Rich vehicle routing problems: From a taxonomy to a definition," *European Journal of Operational Research*, vol. 241, no. 1, pp. 1–14, 2015.

[5] "Real world optimisation with life-long learning." `http://www.rollproject.org`.

[6] J. Mendoza, C. Gueret, M. Hoskins, H. Lobit, V. Pillac, T. Vidal, and D. Vigo, “Vrp-rep: the vehicle routing community repository,” in *Third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog)*, Oslo, Norway, 2014.

[7] J. Mendoza, L.-M. Rousseau, and J. Villegas, “A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints,” *Journal of Heuristics*, vol. 1, pp. 1–28, 2015.

[8] NEO, “Vrp instances.” `http://neo.lcc.uma.es/vrp/vrp-instances/`.

[9] D. Goeke and M. Schneider, “Routing a mixed fleet of electric and conventional vehicles,” *European Journal of Operational Research*, vol. 245, no. 1, pp. 81 – 99, 2015.

[10] GraphHopper. `https://graphhopper.com`.

[11] “Road haulage association.” `http://www.rha.uk.net`.

[12] “Carbon earth.” `http://www.carbonearth.co.uk/carbon-offset.asp`.